EXTENSIBLY MARKED-UP

Robert Filman • Lockheed Martin • filman@computer.org

This month the Spider tours the Web looking to learn about the Extensible Markup Language—the latest latest greatest thing.

Markup languages incorporate markup instructions into a stream of text. These instructions serve two primary purposes: to indicate the structure of a document and to guide the document's presentation. For conventional text documents, structure translates into things like chapters, sections, and paragraphs; presentation, into things like color, font, and type size. However, if you read "document" as "data," then structure can become the rows of a database table, composite objects, or the elements of a parse tree; and presentation can be alternative graphical user interfaces, selected document subsets, or particular document views.

HTML is a simple (at least it used to be simple) instance of a markup language. By putting commands like "<A HREF="http://. . .">See this</A>" into a text stream, an HTML viewer knows to display the text "See this" as a link and to treat the activation of that link as a request to display the page at "http://. . . ." HTML is also a limited markup language. A user can't meaningfully extend the set of HTML tags and is restricted to the behaviors and presentations understood by the target browsers.

HTML is an instance of Standard Generalized Markup Language, a metalanguage for describing markup languages. That is, HTML is a particular selection of tags and meanings of tags within the framework defined by SGML. (SGML is what introduced all the angle brackets and slashes.) In HTML, the markup information is primarily about presentation. Only a few commands—such as headings and paragraphs—convey structure.

However, markup can also be used to elaborate more highly structured information for both display and further processing. XML provides user-definable notions of structure and presentation. Just as different applications need different database schemata, XML can describe a variety of structures. This allows databases of information to be sent out as Web pages. (In the penultimate year of the second millennium, this is an ironic contrast to traditional database concerns with representational compactness, as in, "I can't waste the space to repeat '19' all over the place." The XML standard favors human readability over compactness.) XML is heralded as HTML's replacement, and as a simplification of SGML.

The Spider set out to learn about XML. He found a lot out there.

Xml.com • www.xml.com/

Seybold Publications and
O'Reilly & Associates

All I ever wanted to know about XML I found at xml.com.

The site is a compendium of XML news and documentation. I particularly liked Tim Bray's Annotated XML Specification (http://www.xml.com/axml/testaxml.htm). (Bray is one of XML's authors.) The Annotated Specification page presents a two-panel view, with the specification in the left pane sprinkled with large annotation marks. Clicking on one of the marks displays the corresponding text in the right pane. Annotation marks indicate the marks' flavor, including historical notes, technical amplifications, advice on how to use constructs, and examples. There are also some pages devoted to how Bray built the annotated reference. Ironically, since the current set of browsers do not do XML, his program generates HTML. Presumably, an XML browser would not have needed such preprocessing.

Also on the site is the text of a *World Wide Web Journal* issue devoted to XML (http://www.xml.com/xml/pub/97/10/02/). The issue contains 18 articles, and includes introductions to XML and XML processing, histories of XML and its evolution from other Web data formats, and discussions of tools, style, and the application of XML in various domains. The Bosak article discussed below is included in the issue.

XML Essays • members.aol.com/simonstl/xml

Simon St. Laurent

Trying to figure out where all this XML stuff fits in? Simon St. Laurent's small (well, three) collection of essays on XML gives a balanced presentation. As St. Laurent points out, "XML is new, it's exciting, and it's got to be good, because the specification for it looks indecipherable." He further notes, "Many developers are wondering why exactly they need to learn yet another language." St. Laurent's key point is that XML is a standard for structured information exchange. Web pages are just one

The Spider's rating system:

🪰 🪰 🪰 🪰 🪰 5 Flies = Scrumptious

🪰 🪰 🪰 🪰 4 Flies = Tasty

🪰 🪰 🪰 3 Flies = Filling

🪰 🪰 2 Flies = Edible

🪰 1 Fly = Yuck!

example of such an exchange. XML's prime virtue over SGML is that it's not SGML. SGML has a horrible reputation, and calling the new thing something else gives it a fresh start.

St. Laurent's first essay is a good place to turn for an overview of the concepts and motivations for XML, without getting into the details of the syntax. 🪰 🪰 🪰

XML, Java, and the Future of the Web • metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm

Jon Bosak, Sun Microsystems

Bosak's essay is the best technical presentation I've found on the benefits of XML. He presents four scenarios for XML use:

- Mediation between heterogeneous databases (for example, allowing medical information systems to exchange patient data, eliminating data re-entry and allowing applications to quickly note critical issues like drug allergies).
- Common data representations for use by client-based manipulation tools (for example, design tools in the semiconductor industry). Bosak characterizes this as "XML gives Java something to do."
- Presentations customized to user viewpoints (for example, showing only the sections of a multicomponent manual that apply to a customer's system, or presenting one manual to a user and a different one to a system administrator).
- Agent-based systems, where a particular user's algorithms and preferences can be applied to the data (for example, customized TV guides that list only "interesting" shows).

Bosak also notes XML's ability to present alternative varieties of links (such as location-independent naming, bidirectional links, attributes on links, and so forth) and multiple, computationally rich presentations. 🪰 🪰 🪰 🪰 🪰

XML Activity • www.w3.org/XML/Activity.html

Dan Connolly and Tim Bray

Web activity seems to have a fondness for acronyms that rivals the DoD (excuse me, the Department of Defense.) To clarify the alphabet swirl around XML, the Spider turned to the W3C (World Wide Web Consortium) page on what's happening with respect to XML. There I found descriptions of RDF (the Resource Description Format), which uses XML as a metalanguage for expressing properties of elements such as documents and images and their relationships; PICS (Platform for Internet Content Selection), a way of labeling material so that teenagers have an easier time searching for it; SMIL (Synchronized Multimedia Integration Language), an XML application that provides scheduled Web presentations; and DOM (Document Object Model), a set of classes and methods for manipulating XML content.

I also found that W3C has working groups on schemas (defining the structure and content of XML documents); links (richer hypertext links for XML, including the XML Linking Language and XML Pointer Language); information sets (which seek to make XML descriptions more abstract); fragments (transmitting part of an XML structure); syntax (style sheet mechanisms for presentation, low-level syntax concerns, and so on); as well as a coordination working group, to try to limit the entropy induced by having so many working groups. 🪰 🪰 🪰

Extensible Markup Language (XML) software index • www.xmlsoftware.com/

James Tauber

Both the W3C and the other standards group concerned with XML (the Graphic Communications Association, the parent organization for SGML; http://www.gca.org/conf/xml/xml_what.htm) list XML resources, including programs for manipulating XML documents. I found a more comprehensive list of XML software at Tauber's site. Tauber organizes his collection into the following categories: editors, parsers, browsers, DTD tools, toolkits, Xlink tools, style editors, database systems, serialization, publishing, and conversion tools. A good place to start if you have XML programming to do.

🪰 🪰 🪰 🪰

About the Spider

The Arachnoid Tourist scours the Net to find and review Web sites of interest to IC's readers. What makes a site interesting? Noteworthy sites offer useful technical information, provide tools that can be used in the engineering process, or illustrate how to develop better Internet applications.

Contact Robert Filman at filman@computer.org or the magazine at internet-computing@computer.org.